

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
1. Abend	Vertiefen des Klassen-Konzept in Python	<ul style="list-style-type: none"> • Repetition class(), methoden, instance-variablen, properties • private / public • Statische Methoden • docStrings • Vererbung • Reflaction (Zugriff auf docStrings, dynamisch Function-Call) • Enumerations (<i>Good morning. Pls show me some examples for enums in Python</i>) 	Problem-Based	3h 50'	Code-Refactoring
	Refactoring Logger-Klasse	<ul style="list-style-type: none"> • Requirements überarbeiten • Interface (Abwärtskompatible) überarbeite (Cleancode) <ul style="list-style-type: none"> ○ Public/private/Properties/setter/getter ○ Naming ○ ENUM für Log-Level ○ docStrings / API Doc <p>⇒ Refactoring Weather Application (Usage of Logger-Klasse)</p>	Problem-Based	3h 50'	Code-Refactoring
2. Abend	Umgang mit Modulen und Packages	Leistungskontrolle 1 (Design and Implement a Class) <ul style="list-style-type: none"> • Unterschied Packages / Module • Dokumentation von Modulen • Namespaces • Selektiver Import 	Problem-Based	1h 2h 50'	Code-Refactoring

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
3. Abend	Exceptionhandling in Python	<ul style="list-style-type: none"> • Abfangen von Exceptions: Try / except / else / finally • Exception werfen: Raise • Exception-Handling: lösen/werfen • Eigene Exceptions 	Problem-Based	3h 50'	Code-Refactoring
4. Abend	Multithreading in Python	<ul style="list-style-type: none"> • Threads kreieren (fork) / stoppen • Kommunikation zwischen Threads • Thread-Synchronisation • Locks • I/O Interrupts • Time-Interrupts 	Problem-Based	3h 50'	Code-Refactoring
5. Abend	Hardware ansteuern	Leistungskontrolle 2 (Design and Implement a Sub-Class) <ul style="list-style-type: none"> • GPIO (LED-Steuerung) • PiPlates <ul style="list-style-type: none"> ◦ Relais-Karte ◦ Analog/digital I/O <p>⇒ Mutteruhr für SBB-Uhr ⇒ Selecta-Automat</p>	Problem-Based	1h 2h 50'	Code-Refactoring
6. Abend	Template Mechanismus	<ul style="list-style-type: none"> • Template-Mechanism (Jinja2) <ul style="list-style-type: none"> ◦ Give me an example for JINJA Templates ◦ Can I also implement loops and if-then-else in templates? 	Problem-Based	3h 50'	Code-Refactoring

Unterrichtsplan

Walter Rothlin

7. Abend	Web-Anwendung mit FLASK	<ul style="list-style-type: none">• Hello-Word with FLASK• Request / Responses• Static HTML-content• Mimetype und JSON responses• Parameter Übergabe (get/put)• Endpoints	Problem-Based	3h 50'	Code-Refactoring
8. Abend	MLZ	Modullernzielkontrolle (MLZ)	Ein lauffähige Applikation Nach Vorgaben unter Zeitdruck implementieren.	4h	
9. Abend	Fachgespräche	Einzelne Fachgespräche über MLZ Selbständiges Arbeiten an: <ul style="list-style-type: none">• XML processing in Python (XPath, Schemas, XSLT)• Excel read/write access• RegEx		3h 50'	

- Bemerkungen:**
- Jeder Abend dauert 4 Lektionen.
 - Der Unterrichtsplan kann bei Bedarf dem vorhandenen Wissen der Klasse angepasst werden.
 - Die Studierenden lösen die Übungen auf ihren privaten Notebooks und dem eigenen Raspberry.
 - Der Leistungsnachweis am 8.Aband ist in Einzelarbeit in der vorgegebenen Zeit zu erstellen